

Agentes Autónomos e Sistemas Multi-Agente (AASM)

Ano Lectivo 2007/2008

Laboratório 3 – Agentes Deliberativos no cenário Loading Docks em NetLogo

1 Objectivos

- Implementar o comportamento dos robots no cenário *loading docks* usando uma abordagem deliberativa.

2 Exercício

Usando como base o ficheiro fornecido para esta aula pede-se ao aluno que implemente um agente deliberativo baseado numa arquitectura BDI (*crenças, desejos e intenções*).

O Cap.4 de [Wooldridge02] apresenta a revisão teórica necessária para a compreensão desta aula.

Ao aluno é fornecido um ficheiro com uma implementação do esqueleto de um agente deliberativo caracterizado do seguinte modo:

▪**Ciclo de decisão** – O esqueleto do ciclo de decisão dos agentes está já implementado. Faltam, contudo, a implementação de funções chave que constituem a essência deste laboratório. Assim, inicialmente o agente executa o ciclo de decisão de um agente reactivo, de acordo com o que foi estudado nos laboratórios anteriores;

▪**Crenças** – Os agentes guardam no seu estado interno um mapa do armazém e informação sobre a rampa e as prateleiras que vão encontrando. Adicionalmente, os agentes têm a capacidade de actualizar as crenças à medida que navegam no mapa;

▪**Desejos** – O agente, em cada instante, possui um desejo. Existem três tipos de desejos:

- *apanhar* – o robot deseja apanhar uma caixa;
- *largar* – o robot deseja largar na prateleira certa a caixa que carrega;
- *voltar-inicio* – o robot acredita que as caixas estão todas arrumadas e deseja voltar à sua posição inicial;

▪**Intenções** – As intenções são baseadas nos desejos e determinam o comportamento dos agentes. Essencialmente, desejos são convertidos em intenções que, por sua vez, conduzem à geração de planos. Embora sejam fornecidas primitivas para manipulação de intenções, a conversão de desejos para intenções não está implementada. Uma intenção é um triplo definido do seguinte modo: *<desejo, posição, orientação>*. O 'desejo' refere-se ao desejo associado à intenção. A 'posição' e 'orientação' referem-se a uma posição e orientação que permitem satisfazer o desejo em questão. Por exemplo, uma posição e orientação tal que à sua frente se encontre uma caixa que permita satisfazer o desejo de 'apanhar'.

▪ **Planos** – Os agentes tem a capacidade de executar planos. Um plano não é mais do que uma sequência de instruções (conforme descritas abaixo). A geração de planos não está implementada, embora o código permita a sua execução uma vez definidos;

▪ **Instruções (de planos)** – São suportadas quatro tipos de instruções:

- *instrução-procurar-posição-adjacente* – Indica ao agente que se deve deslocar para uma das posições *adjacentes* (excluindo diagonais) à posição actual do agente;
- *instrução-procurar-orientação* – Indica ao agente para orientar-se segundo um determinado ângulo;
- *instrução-apanhar* – Indica ao agente para tentar apanhar uma caixa, caso esta exista à sua frente;
- *instrução-largar* – Indica ao agente para tentar largar a caixa, se estiver a carregar uma e estiver em frente a uma prateleira vazia da mesma cor;

▪ **Comunicação entre agentes** – O código fornece primitivas de suporte à comunicação entre agentes:

- *envia-mensagem-robot* – que permite enviar uma mensagem a um robot específico;
- *distribui-mensagem* – que envia a mensagem a todos os robots, incluindo o próprio;
- *nova-mensagem* – que trata uma mensagem recebida pelo agente

▪ **Funções de suporte** – Para além das funções para manipular vários tipos de dados (intenções, instruções, planos, etc.) são fornecidas funções que poderão vir a ser úteis (ex: *constroi-plano-caminho*, *posicao-adjacente-prateleira-livre*, *posicao-adjacente-rampa-ocupada*, etc.).

Depois de estudar o ficheiro base, resolva as seguintes questões:

1. Nos laboratórios anteriores, aprendeu que um agente reactivo embora consiga arrumar as caixas nas prateleiras, não sabe quando deve regressar à posição inicial, dado que não tem noção de espaço, nem quantas caixas já foram arrumadas. Agora que os agentes já podem manter estado interno encontre uma solução *eficiente* para fazê-los retornar à posição inicial uma vez arrumadas as caixas. Assuma que os agentes, à partida, sabem quantas caixas têm de arrumar. Essa informação está guardada na variável *NUM_CAIXAS*.

Nota: Não pode chamar directamente a função *caixas-arrumadas*. (Porquê?)

2. Implemente o agente deliberativo baseado em crenças, desejos e intenções. Para o efeito deve implementar o código referente a duas funções: *opções*, *filtra e constroi-plano-com-base-numa-intencao*. A primeira determina o desejo actual, a segunda converte o desejo actual do agente numa intenção concreta. A terceira gera um plano que permite atingir a intenção.

3. Torne as intenções persistentes. O agente não deve desistir de uma intenção assim que esta falhe, devendo procurar alternativas à sua concretização. Por exemplo, durante a execução dos planos associados, se dois robots colidirem, deverão ambos encontrar alternativas para a satisfação das suas intenções. Use a função *colidi* que é chamada quando um agente colide com outro.
4. Discuta alguns dos problemas dos agentes deliberativos desenvolvidos.

Referência: [Wooldridge02] - Wooldridge, M.; *An Introduction to Multiagent Systems*; John Wiley & Sons, Ltd; 2002